CERTIFICATE

I, Tuulikki Tulivirta, hereby certify that, to the best of my knowledge and belief, the following is a true translation, for which I accept responsibility, of Finnish Patent Application No. 20021204 filed on 20 June 2002.

Tampere, 21 May 2003

VIRALLINEN KÄÄNTÄJÄ
TUULIKKI
TULIVIRTA
CERTIFIED TRANSLATOR

Tuulikki Tulivirta
Certified Translator (Act 1148/88)

Tampereen Patenttitoimisto Oy
Hermiankatu 12 B
FIN-33720 TAMPERE
Finland

A method and a system for performing application sessions in an electronic device, and an electronic device

The present invention relates to a system comprising means for performing application sessions in an electronic device with one or more processors and means for alternating resource allocations and the processing of application sessions being run substantially at the same time. The invention also relates to a method for performing application sessions in an electronic device in which the operation is controlled by one or more processors, and in which method resource allocations and the processing of application sessions being run substantially at the same time are alternated. Furthermore, the invention relates to an electronic device comprising means for performing application sessions, one or more processors, and means for alternating resource allocations as well as the processing of application sessions being run substantially at the same time.

As the facilities of wireless communication devices are increased, there is a need to develop systems for utilizing these facilities in a sensible and versatile way. In modern wireless communication devices, it is even possible to perform several application sessions with different purposes, such as a calender, a timer, a telephone directory, a notepad, *etc.* It may often be necessary to run several such sessions substantially simultaneously, wherein problems may occur, for example, when the different sessions would require the same resources of the wireless communication device itself or resources available via connections, at the same time. For example, operating systems are known which are intended for data processors, such as UNIX®, implementing a multi-run method, *i.e.* the processing of several sessions substantially simultaneously. Such a multi-run method is implemented, for example, in such a way that each session is implemented in the form of a single program process and the operating system allocates processing time for each program process in a given order. Thus, when one session is being processed, the other sessions are in a waiting state. Consequently, in practice, the sessions are not processed simultaneously but one after the other. However, the processing time to be allocated for

each session at a time is so short that there is an impression about all the sessions being processed simultaneously. In such an arrangement, none of the sessions must normally have to wait for so a long time that the processing of the session would seem to have interrupted. However, in a situation in which the number of sessions to be processed simultaneously is increased, less and less processing time is left for each session, which will decelerate the operation of all the sessions. In some systems, it is possible to divide the sessions in an order of importance (prioritization), wherein the sessions are allocated processing time according to their priority. In this arrangement, the processing of less important sessions may be decelerated to a significant degree.

Multi-run systems of prior art have also involved the problem that if any session requires a hardware resource and/or an operating system resource, this session will reserve the resource for as long a time as is required for using the resource. Thus, other such sessions which would need the same resource must wait for the release of said resource, because there is normally no such method by which the resource could be transferred from one session to another in such a way that this resource availability situation would be known to each session. For such situations, some solutions have been developed, in which the operating system uses semaphores, or the like, to prevent the use of an allocated resource from other sessions requesting the same.

The processing of several sessions in wireless communication devices should be as close to real time as possible, for example, in view of the convenience of use. For example, when starting a call, a telephone memo session should be run as quickly as possible so that the user of the wireless communication device would find a searched telephone number without an inconvenient delay. However, this may be difficult to implement in real time, particularly when there is a large number of sessions to be processed simultaneously in the wireless communication device. One known solution to meet the demands of multi-run in real time is to use several processors in the same device. Thus, different sessions can be processed in different processors. However, this involves *e.g.* the drawback that when the number of processors is

increased, the power consumption of the device is also increased, which should be avoided particularly in portable devices. Furthermore, in systems of several processors, arrangements are needed, whereby different sessions are allocated to be run by different processors.

For embedded systems, multi-run systems have been developed, in which the processing of different sessions can be scheduled, and each session can be allocated processing time when necessary. However, such embedded systems are designed for specific uses only, wherein it is largely possible to determine in advance, which sessions are to be processed in different situations and which resources will be needed at a time. Thus, the scheduling of the sessions can be performed in advance. However, these systems do not include an operating system. Such systems are not suitable for use in wireless communication devices, in which it is not nearly possible to determine, in advance, all the different use situations and the resources needed in them.

Considerable problems in multi-run systems include the management and scheduling of resources as well as the management and scheduling of application sessions, in other words, the synchronization of the allocation of resources of the processor or of other type, to meet the demands of different sessions. However, these problems cannot be solved separately, wherein it is difficult to find solutions suitable for both problems.

It is an aim of the present invention to provide an improved arrangement particularly for wireless communication devices to implement a multi-run system which operates as close to real time as possible. The invention is based on the idea of forming a session processing environment with different functional units for controlling the processing of sessions and the use of resources. The essential functional units are the functions for controlling application sessions and resources, which are divided into *e.g.* an application session management and scheduling block and a resource management and allocation block. The session to be processed is divided into successive and parallel activities which are implemented in corresponding application blocks. An application and scheduling manager is responsible for the scheduling

(initiating) of the activities of sessions at a time, *e.g.* on the basis of the resources available. A resource allocation manager is responsible for all the centralized management and allocation functions related to the different resource types and resources, for example, by keeping a record on allocated resources and informing the application and scheduling manager of them. An essential feature in the operation of the system according to the present invention is the scheduling between the application and scheduling manager and the resource allocation manager, which is implemented in such a way that after the processing of the resource allocation manager, when the allocation situation of at least some resources is stable and as up-to-date as possible, operations of the application and scheduling manager are processed substantially immediately. Other functional units include, for example, resource handlers for each resource type, as well as application block containers. The resource handlers maintain a record of allocations of the respective resource types in their resource instance tables. Each application block container preferably contains such application blocks needed by one or more sessions, whose processing is normally successive without a need for simultaneous processing. The resource handlers may be preferably implemented as software processes whose priorities are higher than the priority of the resource allocation manager, wherein the resource allocation manager receives **[?]** from the operating system to the processor in a situation when the resource handlers have processed the messages from the applications and the messages from the resources, and the resource allocation situation is temporarily unambiguously known, at least for some resources. Another functional feature of the invention is a session control protocol which is used *e.g.* in the transmission of messages between different parts of the system.

To put it more precisely, the system according to the present invention is primarily characterized in that the application session to be processed comprises one or more application blocks in one or more application block containers, and an order of processing is defined for said application blocks; that the system comprises resource type specific resource handlers for allocating resources for an application session, resource allocation means for investigating and storing the resource

allocation situation, application and scheduling means for selecting at least the application session next in turn for processing on the basis of said allocation situation, processing means for selecting and processing the application block in turn for processing in the selected application session; and the system is provided with a communication protocol between said resource handlers, resource allocation means, application and scheduling means, and processing means, for determining the processing order and for implementing data transmission between said allocation means, resource allocation means, application and scheduling means, and processing means.

The method according to the present invention is primarily characterized in that the application session to be processed comprises one or more processing blocks in one or more application block containers, and an order of processing is defined for said application blocks; that the method comprises at least the following steps:

- a resource management and allocation step for allocating resources for an application session,
- an investigation and storage step for investigating and storing the resource allocation situation,
- a selection step for selecting the application session next in turn for processing, at least on the basis of said allocation situation,
- a processing step for selecting and processing the application block in turn for processing in the selected application session,

wherein in the method, the communication protocol between said resource management and allocation step, investigation and storage step, and the processing step are used for determining the order of processing and, if necessary, for transmitting data between said resource management and allocation step, investigation and storage step, selection step, and processing step.

Furthermore, the electronic device according to the present invention is primarily characterized in that the application session to be processed comprises one or more application blocks in one or more application block containers, and an order of processing is defined for said application blocks; that the system comprises resource type specific resource handlers for allocating resources for an application session,

resource allocation means for investigating and storing the resource allocation situation, application and scheduling means for selecting at least the application session next in turn for processing on the basis of said allocation situation, processing means for selecting and processing the application block in turn for processing in the selected application session; and the electronic device is provided with a communication protocol between said resource handlers, resource allocation means, application and scheduling means, and processing means, for determining the processing order and for implementing data transmission between said allocation means, resource allocation means, application and scheduling means, and processing means.

The present invention shows remarkable advantages over solutions of prior art. By the method of the invention, one session does not necessarily reserve a resource totally for itself and not necessarily for the time of the whole session, wherein it is possible to allocate resources substantially simultaneously to more than one session and to synchronize the resource allocation times required simultaneously by the session as advantageously as possible. The utilization of the resources and the processor in the processing of the sessions can be efficiently implemented in the system according to the invention, presuming that the application and scheduling manager is provided with the sufficient control intelligence. By the method according to the invention, it is possible to control overload situations and to avoid the occurrence of deadlocks. In overload situations, it is possible to suspend application sessions under processing according to the order of priority, and to delay the initiation of new application sessions. This applies to overload situations of other shared resources as well, and not only overload situations of the processor. According to the invention, by the session control protocol and by the principle of prioritizing software processes processing different operations, it is also possible to manage the scheduling of operations in processor overload situations, wherein several software processes may comprise several unprocessed messages at the same time. The order of processing messages which are unprocessed at the same time is determined according to the principle of priority of the software processes participating in the application session within the normal mode of the operating system. Furthermore, the

arrangement of the invention has the advantage that the power consumption of the wireless communication device can be reduced in situations in which the loading situation is so low that the processor can operate at reduced power. The application and scheduling manager makes it possible to detect different loading situations in view of the power consumption for the control. The software architecture according to the invention offers a uniform software environment, wherein the drawing up of application blocks for the wireless communication device is as consistent as possible, irrespective of how complex or simple the session and the respective activity is. Furthermore, the session block structure makes it possible that the programming work needed for processing the sessions can be easily implemented by dividing the session into separate activities, each forming one application block. The software design pattern offered by the block containers can be used for the transmission and reception of the control messages of all the application sessions without a need that the application designer should be aware of the existence of the session control protocol, the application and scheduling manager, and the resource allocation manager. In software development based on application blocks of software architecture according to the invention, it is possible to utilize efficient software development tools and to efficiently re-use both the application blocks and the functional structure combining them into a session profile. In the architecture according to the invention, the application and scheduling manager is responsible for the appropriate initiation of each application block.

In the following, the invention will be described in more detail with reference to the appended drawings, in which

Fig. 1 shows a wireless communication device according to a preferred embodiment of the invention in a simplified block chart,

Fig. 2 shows a system according to a preferred embodiment of the invention in a reduced manner,

Fig. 3 shows the structure and operation of an application block container implemented in the system according to an advantageous embodiment of the invention in a reduced manner, and

Fig. 4 shows the operation of the method according to the invention in a situation of an example session.

In the following, the invention will be described in such respects that the operations needed for implementing the invention will be disclosed. However, even though the different operations of the system will be described below with respect to block structures, it will be obvious that several different blocks participate in the implementation of various activities in cooperation by means of the session control protocol.

Figure 1 shows, in a reduced block chart, a wireless communication device 1 complying with a preferred embodiment of the invention. The wireless communication device 1 comprises a control block 5 with preferably a processor 2 (MCU) for controlling the operations of the wireless communication device 1, for performing operating system functions, *etc.* Furthermore, the control block 5 of the wireless communication device 1 may have a digital signal processor 3 (DSP) for signal processing functions. The processor 2 and the digital signal processor 3 are also provided with a memory 4 for storing data needed in the operation of the operating system, the program code, *etc.* The wireless communication device 1 also comprises communication means 6, such as means for performing mobile station functions. The communication between the user and the wireless communication device 1 may take place via a user interface 7 which preferably comprises a display 8, a keypad 9, and audio means 10a, 10b, 10c. It will be obvious that the above-presented wireless communication device 1 is only a reduced example, but in practice, the wireless communication device 1 may also comprise other functional blocks than those presented above. Also, the digital signal processor 3 is not necessarily needed in the wireless communication device according to the invention, wherein the necessary signal processing functions are implemented in other blocks. Furthermore, the wireless communication device 1 comprises a clock

circuit 11 or the like, for example to generate the clock signals required in the operation of the control block.

Figure 2 shows functional blocks in a system according to a preferred embodiment of the invention in a reduced manner. This system constitutes a kind of a control architecture for controlling the operation of software processes participating in the processing of sessions and the use of resources in the wireless terminal. A software process refers to such a software part which can be scheduled independently in view of the scheduling function processed by the operating system. There are several such resources and resource types for different uses. Some examples of these resources should be briefly mentioned herein: the different functions of the user interface 7, such as displaying of data on the display 8, reading of data from the keypad 9, audio signal processing functions; data transmission functions, such as the reception and generation of various messages and the generation of response messages; signal processing functions; protocol software stacks; other application sessions. In the system, the memory 4 also comprises, for example, a first message buffer B1 for all the messages to be transmitted from resource handlers RH(i), RH(j), RH(k), *etc.* to the application and scheduling manager ASM, as well as a second message buffer B2 particularly for the storage of session initiation requests. Furthermore, the memory 4 of the system is provided, *e.g.,* with a memory area for the storage of a resource allocation table RAT. It will be obvious that in practical applications, also other (dynamic and/or fixed) memory areas can be allocated for different purposes in the memory 4.

The operation of the system can be divided into three basic functions: access control, suspension control, and resource reservation control. The access control is used to restrict the initiation of first application block containers included in the processing of sessions, taking into account the loading situation and the resource allocation situation at the time. The suspension control is used to control the suspension of processing of sessions at terminal points of single application blocks temporarily or finally according to the resource allocation situation. The resource reservation control is used to maintain the resources needed

and requested by application blocks included in the sessions and the resource reservations valid at the time, as well as to transmit them to the application and scheduling functions.

To implement the above-mentioned basic functions, the system is provided with some functional units. The essential functional units of the system are the functions for controlling application and resource management MG, which are divided., *e.g.,* into an application and scheduling manager ASM and a resource allocation manager RAM. The application and scheduling manager ASM is responsible, *e.g.,* for allowing and scheduling the initiation of sessions, scheduling the initiation of processing of activities included in startd sessions, and similar functions independent of applications. A more detailed description of the functions of this block and also the other blocks in the system will be presented below in this specification. For the purpose of resource management and allocation bookkeeping, the system is provided with resource-specific resource handlers RH(i) as well as a resource allocation manager RAM. The activities included in the application sessions are implemented in application blocks which are placed, in a way appropriate for the capacity of the system, in application block containers AC, each of which may contain application blocks AB included in one or more sessions. A single application block AB may be included in different sessions when processed at different times. Furthermore, a session control protocol (SCP) is developed for the system to control the initiation of the processing of session application blocks, as will be described below in this specification. Although the system is shown as separate blocks in Fig. 2, each block is, in practice, preferably implemented as program codes of the processor 2, comprising the program steps needed in the operation of the blocks. The processing of the system is controlled by means of a scheduler of the operating system, or the like, which allocates processing time for the software processes which include the program codes of the system, according to the priority set for them.

The term session (application session) used in this specification must not be interpreted in a restricted manner, but the session may be, for example, a simple set of program commands intended for a single

activity and forming a single application block, or the session may consist of a set of separate program modules or the like which may be grouped into one or more application blocks. The session may also be dependent on other sessions, for example if one application session starts a session from another application to obtain information needed by it from the latter one. The application session can also be called an application. Non-restrictive examples to be mentioned of sessions to be processed in a wireless communication device include a calendar, a telephone memo, answering an incoming call, writing and transmitting short text messages, receiving and reading short text messages, *etc.* In the present invention, each session is formed in such a way that it comprises one or more activities. The processing order of the application blocks corresponding to these activities may be fixed, or the application and scheduling manager ASM may, during the session, determine which application block is to be processed next. The selection depends, for example, on the final result of the preceding application block or the occurrence of a given event, the arrival of an unexpected signal, *etc.* These application blocks included in the same session may be placed in for example one application block container, if the session does not include activities to be performed simultaneously.

The sessions under processing consist of a set of successive and/or parallel activities, the respective application blocks to be processed being placed in one or more application block containers, depending on the demands of parallelism. For example, in the situation of an incoming call, a set of sessions is being started in succession, to inform about the incoming call, to retrieve, on the basis of the calling number, the corresponding name data from the telephone memo of the wireless terminal, and to display the name of the person on the display of the wireless communication device, and to wait if the user of the wireless communication device answers the incoming call, that is, preferably to remain waiting for keystrokes. If the user presses, for example, the handset key, the keystroke is interpreted to decide on further operations, such as answering of the call. The different operations in the above-mentioned example situation can be implemented as separate sessions or as different activities of the same session. Each of said activities can be understood as a kind of a session. Within one session,

however, the allocation and release of resources needed by the application blocks corresponding to different activities is more flexible than it would be if the activities were separate sessions.

In the following, the primary activities of the essential functional blocks will be described. The application and scheduling manager ASM is responsible for the interlacing (scheduling) of the processing of sessions to be processed simultaneously, and the application blocks included therein, with respect to each other, that is, for the control needed for scheduling the allocation of processing time and other resources required for each application block. The application and scheduling manager ASM is also responsible for the reading of application session profiles from an application profile table APT. The application profile table APT contains information, a kind of an activity diagram, about the mutual dependencies of succession and parallelism of the successive and parallel application blocks of the session (*i.e.*, activities included in the session). On the basis of this application session profile, the application and scheduling manager ASM knows how the processing of the session shall proceed from one application block to another. For the same application, it is also possible to draw up more than one application session profile, for example, for two different use situations, for example for working hours and leisure time. It should be noted that the same aplication block may be included in several different application session profiles. For example, it is possible to start several sessions of the same application at different times, each session profile being different with respect to some activities. Thus, blocks included in the same application block container can thus be processed in different sessions without a need to copy these blocks for each session. This has, for example, the advantage of needing less space of the memory 4. Preferably in connection with the start-up, the application sessions are allocated *e.g.* a priority whereby it is possible to control the order of processing and the possible suspension of activities of sessions to be processed simultaneously. Furthermore, the estimated processing time and the resources required by each application block are preferably defined for each application block of a session. This data can be used by the application and scheduling manager ASM in the control of scheduling the application blocks and in the estimation of the

resource situation, particularly in overload situations. The application and scheduling manager ASM has access to a block assignment table BAT for the application blocks AB, containing information about the application blocks included in each application block container and the blocks being actively processed in each session. This data is stored in block assignment records BAR.

Furthermore, the application and scheduling manager ASM may have access to information about the resources needed by each application block, either in a fixed or conditional manner, in the application profile table APT containing application profile records APR. If the data of all the required resources is not in the application session profile, they are obtained when the application blocks transmit allocation requests to the resource handlers and to the resource allocation manager.

The application and scheduling manager ASM preferably stores history data about each session. For this purpose, the system is preferably provided with a session history table SHT which comprises a session history record SHR where the application and scheduling manager ASM maintains history data of the session being processed, which is preferably available for the transfer of information output from each application block to the next application blocks.

The application and scheduling manager ASM receives the initiation request messages to be transmitted in connection with new session requests so that the application and scheduling manager ASM can check the loading situation and also take care of the scheduling of the initiation of new activities included in the session to be started. Particularly in a situation of overloading the processor or other resources, the application and scheduling manager ASM makes the decision whether the session requested to be started can be initiated. To make the decision, the application and scheduling manager ASM preferably uses the processing time estimates of the application blocks and the resource allocation time estimates and possibly other data to examine if the previously started, more important sessions can be continued in real time also after the starting of a new session. With such an arrangement, it is possible to control overload situations and to prevent

their development to a great extent. The application and scheduling manager ASM also takes care of the temporary suspension of a session at the terminal point of a processed application block, for example, when the same resource is needed for the processing of an activity in a more urgent session or when a required resource is totally engaged to finish the application blocks under processing. In practice, the suspension of a session can be implemented so that after the end of the previous application block, the application and scheduling manager ASM does not transmit a control command (DO message) to the block container containing the next application block of the same session until it can be allocated processing time. In some situations, the session may comprise application blocks intended to be processed in parallel, which are thus in different application block containers each. If said session suspension point is at the beginning of such application blocks implementing parallel activities, the control command is not transmitted to any of these application blocks until the operation of the session can be continued.

It is also a function of the application and scheduling manager ASM to control messages coming from any resource handlers in a correct order to the correct application blocks in correct application block containers when no application block is synchronically waiting for an incoming message. This is necessary particularly in asynchronic message transmission in a situation, in which the capacity of the same resource is divided to more than one application block or when the corresponding allocation request was transmitted by an application block which does not itself remain waiting for the corresponding allocation notification (In message). In other cases, the message coming from the resource handler might be forwarded to such an application block container or application block, for which the message was not intended. The application and scheduling manager ASM also participates in the releasing of resources by providing the application blocks with the necessary control, for example, in overload and error situations.

The resource handlers keep a list of all the resource reservation instances RI in resource instance records RIR to be stored in corre-

sponding resource instance tables RIT. When an application block contained in an application block container requests for access to a resource, information about the type of the resource need is preferably transmitted from the application block directly to the respective resource handler RH or to the resource allocation manager RAM. For example, in connection with resources formed for data transmission, it may be possible to select the data transmission rate, the error rate required of the data transmission, *etc.*, wherein parameters affecting these properties of the resource reservation instance can be transmitted in the resource request (allocation request Con). Thus, after receiving the resource request, the resource handler RH or the resource allocation manager RAM preferably examines if a resource according to the requested parameters can be allocated for the application block that requested for the resource. A previous resource allocation may also be available to be re-allocated, if it meets the property demands included in the request. The application block or the application block container may use a return message to release a resource instance possessed by it, wherein the resource handler RH marks the resource instance vacant in the sense that another application block or application session may request for the allocation of the resource. If the reservation and/or release of a resource requires that some measures are taken by the resource itself, the resource handler RH is informed of unaccomplished reservation or releasing instances, and the application block or application session requesting for the resource may need to wait for a reservation notification. The resource allocation manager has information about the resource reservation situation of all the resource types. The aim of this arrangement is to prevent the development of situations of overloading any resource.

The application and scheduling means (ASM) and the resource allocation means are preferably formed to have no interspaces, wherein the changes relating to data of the sessions are stored in said session history table (SHT), and the changes relating to data of resource reservation instances are stored in said resource allocation table (RAT). Also, the resource handlers are preferably formed to have no interspaces, wherein they store changes relating to data of the resource reservation instance in the resource instance table (RIT).

Also, more than one resource reservation instances can be made for the same resource, for example, in a situation in which several application blocks together or several sessions have a need to use the same resource substantially simultaneously. These application blocks using the same resource are normally not in the same block container but they can be in different containers and different sessions. In such a situation, it is important to make sure that the resource requests (Con messages) to the resource handler and the allocation notifications (In messages) from the resource handler can be linked together. For this purpose, the resource allocation manager RAM is provided with functions, whereby it processes the resource requests related to the same resource handler, for example, takes care of queueing and transmits the requests at the right time and in the right order to the resource handler. The transmission of allocation notifications from the resource handler to the correct block container, corresponding to the correct resource request, is based on the fact that the application block container that transmitted the request is always active to wait for the allocation notification, and each resource request contains an unambiguous reference identifier. In connection with the transmission of each resource request, the application block provides the request with the reference and informs the application and scheduling manager ASM of the request and its reference, to be stored as a correlation reference record CRR in a correlation reference table CRT. The use of references makes it possible that there are several resource requests simultaneously and that allocation notifications are waited for also asynchronically.

In the same correlation reference table CRT are also stored the references of such application messages, for which the application session is waiting for a reply message via a resource handler (*e.g.* a data transmission connection). The use of references makes it possible that several requests exist simultaneously and are asynchronically waited for.

The application block itself is also responsible for the releasing of the resource reservation instance when the application and scheduling

manager has not instructed the resource reservation instance to be reserved for the next or later application blocks of the same session. In possible error situations, the application and scheduling maanger ASM starts the necessary application-independent release function to be processed. Thus, the resource will not be left unnecessarily reserved.

The actual processing of resource requests takes place in resource handlers, preferably one being formed for each resource type. The resource handler RH(i) is a software process (handler) intended for using reservation, release and other control requests relating to a given resource in the same way in the case of all the different sessions and different resource types. In the resource handler RH(i) formed for each resource type i, the specific features of the type in question, *e.g.* the processing of video signals, audio signals, text messages, *etc.*, are taken into account . The resource handler maintains a list of resource reservation instances formed for using the resource in question. The processing of resource handlers is scheduled according to the need determined by the application sessions, by using a scheduler included in the operating system.

The application block container AC is a software process formed by application blocks included in one or more different application session profiles and by the skeleton frame of the application block container. There are normally several application blocks, and they are processed in succession within the same container but in parallel in the case of different containers. In each application block container, one block is an idling block IB to wait for asynchronic events. Furthermore, each block comprises an entrance state module ESM, a branching state module BSM, and a termination state module TSM for terminating the processing of the block container (*i.e.*, for removing it from the active state). Also, each application block comprises a start state module SSM at the beginning and a stop module SM at the end. The application block container AC also comprises *e.g.* a list of variables needed in the operation of the block container, data structures, memory reservations, information about the resource reservation instances, *etc.* The block container AC receives messages to be transmitted from the application and scheduling manager ASM to start the processing of

application blocks. Furthermore, the above-mentioned modules and the idling block included in the block container transmit return messages (Out messages) of the session control protocol to the resource allocation manager RAM which transmits them further to the application and scheduling manager ASM. These return messages relate, for example, to situations in which the processing of an application block has been suspended to wait for the meeting of a condition, *e.g.* the arrival of an application message in the wireless communication device, or the processing of the block has been terminated.

The session control protocol SCP developed for the system according to the present invention is used, for example, to provide cooperation of the functional blocks in the system on the basis of the transmission of session control messages between the functional blocks. This is a kind of a message transmission function between the functional blocks participating in the processing of the application session. The session control protocol SCP can be used to control the loading of the processor of the wireless communication device to avoid and control overload situations. Furthermore, the session control protocol can be used to control the scheduling of the initiation of single activities during an application session, and to transmit process identifiers PID of software processes belonging to the functional blocks participating in the session, as well as other application-independent data between the functional blocks. The session control protocol can be used to arrange the prioritizations between the application sessions, for example, on the basis of the scheduling of activities included in the sessions. The arrangement of the reservations of shared resources, such as resources related to data transmission connections, in an optimal way is made possible for the application and scheduling manager in the system according to the invention by means of the session control protocol.

The aim is to implement the session control protocol in such a way that the functions related to resource reservation, such as resource requests (Con messages), can be implemented in different application sessions as uniformly as possible, irrespective of the resource type.

This makes it easier, for example, to compile the program codes for the application blocks.

In the session control protocol, various messages are used for controlling the system and for transmitting data between different functional blocks. In this context, these messages are commonly referred to as session control messages, and they include *e.g.* the control command Do, the return message Out, the resource request Con, and the reservation notification In. Also, separate messages are required for exceptional situations which, however, do not need to be discussed in more detail in this context. The data structures may vary in different messages, but each message preferably contains information about the application session it relates to, as well as the program process from which it was transmitted and/or for which it is intended. Also, the messages typically comprise one or more data fields, in which data complying with the message type is transmitted to the receiver of the message. Messages of the session control protocol are used to transmit, for example, identifiers of dynamically generated or selected resource handlers or application block containers (software processes) participating in the same session.

As already stated above, the application block is started by a control command (Do message) transmitted by the application and scheduling manager ASM. The control command can be used to inform the application block of the resources which it can reserve. Furthermore, the control command can be used to inform the application block of the resources which the application block should reserve or release. An example to be mentioned is a situation of setting up a call, the necessary data transmission resources being vacant. Thus, the application block to be used for setting up the call can be informed that it should reserve the data transmission resources required for the call. After the processing of the application block has been completed, it transfers a return message (Out message) to the resource allocation manager RAM. If necessary, this return message comprises *e.g.* information about the resources reserved and/or released by the application block. Furthermore, the return message (Out message) is transmitted to those resource handlers whose resource reservations are released

by the application block either for other application blocks of the same session or for other application sessions. A later application block of the same session can reserve a vacant resource reservation instance for its use by transmitting to the resource handler a return message (Out message), in which the resource handler receives the identifier of the respective application block container as a sign of this re-allocation. Each resource handler has the identifier of only one block container stored at a time, or when this is missing, the identifier of of the application and scheduling manager or another default block corresponding to the resource type. The above-mentioned arrangement makes it possible that each single resource reservation has separate visibility to the application block containers. After the processing of the resource allocation manager RAM, the application and scheduling manager ASM is substantially next in turn for processing because of its high priority; consequently, the situation of using and allocating resources is, at least partly, up-to-date at the moment, for the application and scheduling manager ASM. Thus, the application and scheduling manager ASM can find out which application blocks can be started and which resources can be allocated for them.

As a summary of resource allocation, it should be stated in this context that the application block may have access to a resource directly required by it during the processing of the application block, if the resource is vacant. Thus, the application block transmits a resource request to the resource handler corresponding to the type of the requested resource, which examines if the requested resource can be allocated to the application block that requested it. As a response, the resource handler transmits a reservation notification containing either a positive acknowledgement (ACK) if the resource can be allocated to the application block in question, or a negative acknowledgement (NACK), if the resource cannot be allocated to the application block in question at that time. Presuming that the information for controlling the resource allocation, given in the control command from the application and scheduling manager ASM to the application block, is exact, the reservation notification from the resource handlers is normally not negative. If the need for resources is not necessarily immediate, or the above-mentioned control command tells that the resource is reserved,

e.g., for another use, the application block may transmit a resource request to the resource allocation manager RAM which applies the principle of queueing. At this stage, the application block container whose application block transmitted the resource request to the resource allocation manager RAM will remain waiting for the reservation notification either synchronically or asynchronically. Later on, when allowed by the resource allocation situation, the resource allocation manager RAM transmits the waiting resource request to the respective resource handler which forwards the reservation notification directly to the application block container which requested for the resource. To make this possible, the application and scheduling manager ASM may have to transmit control commands (Do messages), *e.g.*, to application block containers of parallel application sessions, to release some resource reservations.

The functions according to the present invention can be largely implemented by software as program codes in the processor 2 of the wireless communication device 1. The resource allocation manager and the resource handlers can be thought to form form a kind of middleware between the operating system and applications of the wireless communication device 1. From the point of view of the operating system, the operations of the system according to the invention include processing of one or more threads to be scheduled. The invention is not bound to any operating system, but the invention can be applied in connection with such operating systems which have the multi-run facility. However, the operating system is preferably such in which, during multi-run processing, the suspension of the processing of one schedulable program process (thread) and the continuation of the processing of another program process cannot take place at any point of the program code but only at predetermined points. The compiler of the program selects and sets the points in which the operating system can change the software process under processing. Such suspension points are preferably located at points where the processing of the application block container or the whole application session can be suspended by a natural cause, for example upon waiting for a response from a resource handler.

In the following, the operation of the method according to a preferred embodiment of the invention will be described with reference to the example situation shown in Fig. 4. In the figure, the arrows indicated with solid lines illustrate the transmission of messages between blocks, the arrows with broken lines illustrate the procedure of processing an application session from one activity to another, and the arrows with dotted lines illustrate the transmission of control messages of the application session between the blocks.

Let us assume that the wireless communication device 1 is turned on and that the operation system has been started. At some stage, the operating system starts the operation of the system according to the invention, preferably by starting the operations to initialize the application and scheduling manager ASM, the resource allocation manager RAM, and the session control protocol SCP. At the stage when any application session is to be started, an application session request is transmitted to the application and scheduling manager ASM. The initiation of the application session may be caused by a request from another application session, an external event, such as an incoming call to the wireless terminal 1, a selection by the user of the wireless terminal 1, *etc.* In the situation of Fig. 4, a text message has arrived in the wireless terminal 1 and caused the initiation of a new application session. The resource handler RH(1) intended for the text message receiving connection type has transmitted an application session request S1 which has been placed in a first message buffer B1. The application and scheduling manager ASM detects the message arrived in the message buffer B1, reads the data needed for identifying the requested application session from the message, and reads the session profile of the application session to be started from a session profile table APT. At this point, also the history data of the application session to be started are initialized in a session history table SHT. The application session profile ASP discloses, *e.g.*, the activities included in the session and the resource reservations possibly required by them, as well as the estimated processing times possibly defined for the application blocks corresponding to the activities. On the basis of the session profiles of the sessions being processed and suspended, the data in the resource allocation table RAT, and the session profile of the

session to be started, the application and scheduling manager ASM makes an analysis to find out if the session can be started. The application and scheduling manager ASM estimates the increase in the loading of the processor 2, if the session is started. Furthermore, the application and scheduling manager ASM examines the allocation situation of resources needed by the session to be started, and also estimates the loading situation of these resources. If the application and scheduling manager ASM determines that the session can be started, the application and scheduling manager ASM will select an application block container AC which contains the first application block in the session profile of the session in question. The selected application block container AC is loaded in the memory 4 for the processing of the first application block, unless it is already in the memory and vacant for processing. In the example of Fig. 4, the application session to be started contains a set of activities, the corresponding application blocks being the application blocks 0B–11B. The processing of the session is started by initiating the first application block 0B at the stage when the application block container AC which contains this application block and was loaded above comes in turn for processing. As already stated above in this specification, a priority order can be defined for the application sessions, on the basis of which the processing time obtained by the sessions, as well as the processing order of the sessions, can be adjusted.

From the message buffer B1, the application and scheduling manager ASM transfers the arrived application session request S1 to a second message buffer B2 substantially immediately after being allocated processing time by the operating system. At the stage when the processing of a new application session can be started, the application and scheduling manager ASM transmits a control command (Do message) to the selected block container AC containing the first application block 0B of the session. As a result, the processing of the application block 0B is started. In this example, after the processing of the application block 0B has been completed and the corresponding return message (OUT message) has been transmitted to the resource allocation manager RAM, the application and scheduling manager ASM (having received this return message from the resource allocation man-

ager RAM) transmits, on the basis of the application session profile, a control message to three application block containers AC1, AC2, AC3 for the respective three application blocks 1B, 2B, 3B, which are intended for forming connections of different types in parallel. In this example, these connections are of the audio, data and video types, respectively. Each application block 1B, 2B, 3B transmits a resource request (CON message) to the resource handler RH(2), RH(3), RH(4) corresponding to the respective connection type. These resource requests contain information, such as a program process identifier or another identifier of the block container that transmitted the message. By means of this information, the resource handler can return a reservation notification (In message) to the correct block container. The resource handler returns a reservation notification (In message) at the stage when the resource handler has transmitted a resource request (Con message) from the block container to the protocol program and received from it an acknowledgement of the completion of the connection. Thus, the application block in the block container receives either a negative or a positive reservation notification, *i.e.*, information that the resource handler has, within the capacity limits, allocated a resource reservation for the application session and that the resource handler has completed possible other initializing and recording operations necessary for using the resource. Furthermore, it is an aim of the resource handler to transmit application messages received from the corresponding protocol program to that block container participating in the processing of the application session, to whose identifier the resource handler has access at the time. The application session will preferably pass to the wait state at the stage when it, after the transmission of the resource request (Con message) remains waiting for the reservation notification (In message) from the resource handler. Information about this shift to the wait state is transmitted to the resource allocation manager RAM preferably by means of a return message (Out message) in connection with the transmission of each resource request. In the application block 4B of Fig. 4, it is waited that a reservation notification is obtained as a response to all the three resource requests until moving on to the generation of data to be transmitted from the wireless terminal 1 along the different connection types. In the case of a general application session, by using different connection types, an application

session can perform an interactive exchange of application messages by the application blocks 5B, 6B and 7B with the applications at the other ends of the connections, wherein it is advantageous that the application blocks 5B, 6B and 7B are located in three different block containers. In the general case, the application session may contain a much larger number of application blocks to be processed in parallel.

The scheduling of the processing of the resource handlers is arranged by means of the operations and schedulings of the operating system so that the operations of the resource handlers are processed after the application blocks have transmitted messages to them for processing. The resource handlers are provided with an interface, via which the application containers can transmit session control messages and application messages to the resource handlers and, correspondingly, receive messages from the resource handlers. This interface is independent of the application and substantially also of the resource type, wherein it is possible to use messages of substantially standard format as messages for the control protocol of the application session. Furthermore, the resource handler transmits application messages both from the application containers to the resource and from the resource to the application containers and/or the application and scheduling manager ASM. For each resource request, the resource handlers form a resource reservation instance RI which is a kind of a definition of the resource reservation instance as well as the parameters related to the use of the resource, such as the data transmission rate, the resolution of the display, stereo/mono sound, *etc.* Resource reservation instances are stored as resource instance records RIR in a resource instance table RIT. Each resource handler has its own resource instance table. Each resource handler takes care of its own resource instances and transmits information about them to the resource allocation manager RAM. Consequently, in the resource allocation manager RAM, the data about all the allocated resource instances are preferably given in the resource allocation table RAT.

The resource allocation manager RAM transmits the data contained in the resource allocation table to the application and scheduling manager ASM, *e.g.*, to bring the processing of the different application ses-

sions *e.g.* from the suspended state to the processing state. The resource allocation manager RAM also receives the return messages from the application block containers, such as data about the termination of the processing of the application block, or about a synchronic wait state starting in the application block, or about the incoming of a reservation notification from the resource handler in an application block container.

After the resource allocation manager RAM has transmitted the data of the resource allocation table to the application and scheduling manager ASM, the processor 2 gives the processing turn to the application and scheduling manager ASM which will select the next application session and application block in turn for processing. As the grounds in the selection, the application and scheduling manager ASM uses *e.g.* the resource reservation situation and the situation of loading of the processor 2. A control command (Do message) is preferably transmitted to the selected block container containing such an application block to start the processing of the application block. There may also be more than one different application blocks in different block containers in turn for processing (so-called parallel program processes). In practice, this means that the processor 2 alternates the processing time to be allocated to these application block containers. Examples of such parallelism are the application blocks 1B, 2B, 3B and also 5B, 6B, 7B and 8B, 9B, 10B shown in the example of Fig. 4. However, in multiprocessor systems, real simultaneity is also possible for processing application block containers.

Let us assume that the sub-programs 5B, 6B, 7B are in turn for processing, which form the data content of audio, data and video types to be transmitted from the wireless terminal 1, as well as the application messages required for the control of the data transmission. Consequently, these application blocks 5B, 6B, 7B generate the data content to be transferred and transmit the necessary application messages to the respective resource handlers RH(2), RH(3), RH(4). These messages are indicated with the references D, E, F in Fig. 4. The resource handlers take the necessary measures to transmit the data contents of the three different types from the wireless terminal 1. After these three

data transmissions have been made, return messages (Out messages) are generated in the application blocks 8B, 9B, 10B to release the respective three resource allocations. After this, the processing of the application session can be terminated by processing a termination block 11B. The memory spaces allocated for the released block containers AC1, AC2, AC3 can be vacated and the data of the resource allocations used by the terminated session can be deleted from the resource instance tables RIT and the resource allocation table RAT.

The processing of the next application session request S2 is performed by applying the principles presented above.

Further, Fig. 3 shows the structure of an application block container AC and the procedure of the processing of the application blocks in the application block container AC. The application block container has an entrance state module ESM which is always processed for the block container which is vacant when the first control command comes in. After this, the application block container has a branching state module BSM, in which the identifier of the application block to be processed next is examined to find out the application block AB1, AB2, AB3, to whose beginning the processing logic of the block container should be branched. At the terminal end of each application block, there is a stop module SM indicating the termination of the processing of the block to the resource allocation manager RAM by means of a return message (Out message) and thereby further to the application and scheduling manager ASM, after which the processing logic of the block container AC shifts to wait for the next control command (Do message) to be able to continue the processing of the session from the next application block in turn when this block container is allocated processing time by the operating system. For this purpose, the processing logic moves from the stop module SM to the beginning of the idling block IB with an idling state module ISM intended to receive any messages coming from the resource handlers to the application block container. Such a message in the message buffer is received by the idling block, after which the final state module FSM at the end of the idling block transmits a return message (Out message) to the resource allocation manager RAM and further to the application and scheduling

manager ASM, informing about the received message. The processing of the idling block IB is terminated when the processing logic of the block container shifts to the beginning of any application block on the basis of the next control command (Do message) received in the idling state module ISM. Thus, the processing logic shifts from the final state module FSM at the end of the idling block IB via the branching state module BSM to the beginning of the application block to be processed next. The application and scheduling manager ASM can use a particular parameter in the control command to instruct the processing logic to move from the idling block after the final state module FSM to the terminal state module TSM of the application block container, to terminate the processing of the block container and to release the container. After the terminal state module TSM, the application block container can be, in view of the application and scheduling manager ASM, freely selected for the processing of any application block therein, within any application session. When the message received in the idling block IB does not cause shifting of the processing logic to the branching state module BSM or the termination state module TSM, the processing logic will always return from the final state module FSM of the idling block to the idling state module ISM at the beginning of the idling block, to continue the idling.

In situations, in which the loading of the processor 2 is relatively low, the application and scheduling manager ASM can adjust the power consumption of the processor 2 to be lower, *e.g.* by reducing the clock frequency. On the other hand, the power consumption can, in some processors, be controlled by setting at least some of such functional blocks of the processor which are not needed at the moment, in a power saving mode.

The present invention can, to a great extent, be implemented by software, for example as program commands of the processor 2.

It is obvious that the present invention is not limited solely to the above-presented embodiments but it can be modified within the scope of the appended claims.

Claims:

1. A system comprising means (2, 4) for processing application sessions in an electronic device (1) with one or more processors (2), and means (2) for alternating resource instances as well as the processing of application sessions being processed substantially simultaneously, **characterized** in that the application session to be processed comprises one or more application blocks (AB) in one or more application block containers (AC, AC1, AC2, AC3), and an order of processing is defined for said application blocks (AB); that the system comprises resource type specific resource handlers (RH) for allocating resources to the application session, resource allocation means (RAM, RAT, RH, RIT) for examining and storing the resource allocation situation, application and scheduling means (ASM) for selecting at least the application session next in turn on the basis of said allocation situation, processing means (2, ASM) for selecting and processing the application block (AB) in turn for processing in the selected application session, and the system is provided with a protocol (SCP) between the application and scheduling means and the processing means for determining the processing order and for implementing data transmission between said reservation means, resource allocation means, application and scheduling means, and processing means.

2. The system according to claim 1, **characterized** in that it comprises means (RAM, RAT, RH, RIT) for maintaining the resource allocation situation, means (ASM, Do) for transmitting a control command to the application block for transmitting control information on resource allocation to the application block in connection with initiating the application block, and means for transmitting a return message in connection with termination of the processing of the application block to inform about the resource reservation and releasing instances by the application block to maintain the resource allocation situation up to date after the termination of the processing of each application block.

3. The system according to claim 2, **characterized** in that it comprises means (ASM, AC, SCP, RAM, RAT, RH, RIT) for allocating the resources needed by each application block to the application, as well

as for releasing them, on the basis of control data obtained in a control command transmitted from the resource allocation means either directly from the resource type specific resource handlers or from the resource allocation means which make queueing of resource requests possible.

4. The system according to claim 2 or 3, **characterized** in that it comprises means (AC, SCP, RIT, RH) for making the resource instances made by the application session, by means of return messages, available to different application block containers participating in the processing of the session dynamically according to the need.

5. The system according to any of the claims 1 to 4, **characterized** in that the system comprises an operating system with scheduling functions, and that for resource allocation and release and other control by the application and scheduling means together with the resource handlers, a session control protocol is formed by application-independent control messages and rules on their use, which protocol is, in its operation, arranged to implement the scheduling control of the processing of the application and scheduling means, the application block containers, the resource allocation means, as well as the resource handlers, on the basis of priorities defined for the scheduling functions of the operating system as well as for the application and scheduling means, the application block containers, the resource allocation means, and the resource handlers.

6. The system according to any of the claims 1 to 5, **characterized** in that it comprises a resource instance table (RIT) for each resource handler to transmit the resource allocation situation to said resource allocation means (RAM), and that the processing order between the resource allocation means and the resource handlers (RH) is determined so that substantially immediately in turn after the processing of the resource handlers (RH) are the resource allocation means (RAM), wherein the resource allocation situation is, as to the changes occurred, unambiguous in the resource instance table (RIT).

7. The system according to claim 6, **characterized** in that the processing order between the resource allocation means and the application and scheduling means is determined so that substantially immediately in turn after the processing of the resource allocation means are the application and scheduling means (ASM), wherein the resource allocation situation is, as to the changes occurred, unambiguous, and control data aimed at synchronization of the use of various types of resource allocations can be formed in the application and scheduling means, in the control commands transmitted by it.

8. The system according to any of the claims 1 to 7, **characterized** in that one processing stop module (SM) is placed at the end of each application block, an idling state module (ISM) is placed in the application block container containing the application block, and that the processing of the application block container containing the application block is arranged to transmit a return message in the stop module (SM) and to be suspended in the idling state module (ISM) and to wait for a control command from the application and scheduling means, wherein the processing of the application session is suspended in the block container in question.

9. The system according to claim 8, **characterized** in that the application and scheduling means are arranged to analyze the resource allocation situation and the scheduling of the sessions to be processed to detect and control an overload situation of one or more resources by replacing, according to the need, application sessions with application sessions requiring less resources, or by delaying, if necessary, the transmission of control commands to be transmitted to the application sessions, wherein the application session under processing is arranged to be temporarily suspended, or the initiation of a new application session is arranged to be delayed.

10. The system according to any of the claims 1 to 9, **characterized** in that one or more application block containers are formed of said application blocks of the application session, that application blocks in one application block container are arranged to be processed temporally at different times, and that if the application session contains

application blocks intended to be processed substantially at the same time, they are placed in different application block containers.

11. The system according to claim 10, **characterized** in that for forming applications to be processed in the system, each application block container is provided with an interface at points where the processing of an application block or application block container can be suspended and the processing turn can be changed, by means of which interface the transmission and reception of session control messages is arranged to be processed in the application block container without a need to handle messages of the session control protocol when setting up the application.

12. The system according to any of the claims 1 to 11, **characterized** in that the resource handlers (RH) are provided with an interface for transmitting data between the resource handler and the system, the interface being substantially independent of the application session and the resource type.

13. The system according to any of the claims 1 to 12, **characterized** in that it comprises a resource instance table (RIT) arranged for the use of each resource handler (RH), and that the resource handlers are formed to have no interspaces, wherein the resource handlers are arranged to store the changes relating to data of the resource instance in the resource instance table (RIT).

14. The system according to any of the claims 1 to 13, **characterized** in that the application and scheduling means (ASM) comprise a session history table (SHT), and the resource allocation means comprise a resource allocation table (RAT), that the application and scheduling means (ASM) and the resource allocation means are formed to have no interspaces, wherein changes relating to data of the sessions are arranged to be stored in said session history table (SHT), and changes relating to data on resource allocations are arranged to be stored in said resource allocation table (RAT).

15. The system according to any of the claims 1 to 14, **charact rized** in that it comprises means (ASM) for determining the loading situation of the processor and for adjusting the power consumption of the processor on the basis of the loading situation as well as by scheduling of activities of the application sessions.

16. A method for processing application sessions in an electronic device (1), in which the operation is controlled by means of one or more processors (2), and in which method resource allocations as well as the processing of application sessions to be processed substantially simultaneously are alternated, **characterized** in that the application session to be processed comprises one or more application blocks (AB) in one or more application block containers (AC, AC1, AC2, AC3), and a processing order is determined for said application blocks (AB), that the method comprises at least the following steps:

- a resource management and allocation step for allocating resources for an application session,
- an investigation and storage step for investigating and storing the resource allocation situation,
- a selection step for selecting the application session next in turn for processing, at least on the basis of said allocation situation,
- a processing step for selecting and processing the application block (AB) in turn for processing in the selected application session,

wherein in the method, a communication protocol (SCP) between said resource management and allocation step, detecting and storing step, and the processing step is used for determining the order of processing and, if necessary, for data transmission between said resource management and allocation step, detecting and storing step, selection step, and processing step.

17. The method according to claim 16, **characterized** in that the resource allocation situation is maintained, and control commands are transmitted to the application block to transmit control data related to resource allocation in connection with the starting of the application block, and a return message is transmitted from the application block to inform about the resource allocation and release instances by the

application block, to maintain the resource allocation situation up-to-date after the processing of each application block.

18. The method according to claim 17, **characterized** in that resources needed by each application block are allocated to the application and released on the basis of control data obtained either directly from resource type specific resource handlers or in a control command transmitted from resource allocation means which make the queueing of resource requests possible.

19. The method according to claim 17 or 18, **characterized** in that return messages are used to allocate resource reservation instances by the application session to the use of different application block containers participating in the processing of the session, dynamically according to the need.

20. The method according to any of the claims 16 to 19, **characterized** in that in the method, an operating system is used which comprises scheduling functions, and that for resource allocation and release and other control by the application and scheduling means together with the resource handlers, a session control protocol is used, formed by application-independent control messages and rules on their use, which protocol, when in operation, implements the scheduling control of the processing of the application and scheduling means, the application block containers, the resource allocation means, as well as the resource handlers, on the basis of priorities defined for the scheduling functions of the operating system as well as for the application and scheduling means, the application block containers, the resource allocation means, and the resource handlers.

21. The method according to any of the claims 16 to 20, **characterized** in that in the method, a resource instance table (RIT) is used for each resource handler to transmit the resource allocation situation to said resource allocation means (RAM), and that the processing order between the resource allocation means and the resource handlers (RH) is determined so that substantially immediately after the processing of the resource handlers (RH), the resource allocation step is in

turn for processing, wherein the resource allocation situation is, as to the changes occurred, unambiguous in the resource instance table (RIT).

22. The method according to claim 21, **characterized** in that the processing order between the resource allocation step and the processing step is determined so that the processing step is in turn substantially immediately after the processing of the resource allocation step, wherein the resource allocation situation is unambiguous with regard to the changes occurred, and the control data aimed at synchronizing the use of resource reservation instances of various types can be formed at the processing step in the control commands to be transmitted therefrom.

23. The method according to any of the claims 16 to 23, **characterized** in that one processing stop module (SM) is placed at the end of each application block, an idling state module (ISM) is placed in the application block container containing the application block, and that the processing of the application block container containing the application block is arranged to transmit a return message in the stop module (SM) and to be suspended in the idling state module (ISM) and to wait for a control command from the application and scheduling means, wherein the processing of the application session is suspended in the block container in question.

24. The system according to claim 23, **characterized** in that the application and scheduling means are arranged to analyze the resource allocation situation and the scheduling of the sessions to be processed to detect and control an overload situation of one or more resources by replacing, according to the need, application sessions with application sessions requiring less resources, or by delaying, if necessary, the transmission of control commands to be transmitted to the application sessions, wherein the application session under processing is arranged to be temporarily suspended, or the initiation of a new application session is arranged to be delayed.

25. The method according to any of the claims 16 to 24, **charact r-iz d** in that one or more application block containers are formed of said application blocks of the application session, that application blocks in one application block container are arranged to be processed temporally at different times, and that if the application session contains application blocks intended to be processed substantially at the same time, they are placed in different application block containers.

26. The method according to claim 25, **characterized** in that for forming applications to be processed in the system, each application block container is provided with an interface at points where the processing of an application block or application block container can be suspended and the processing turn can be changed, by means of which interface the transmission and reception of session control messages is processed in the application block container without a need to handle messages of the session control protocol when setting up the application.

27. The method according to any of the claims 16 to 26, **characterized** in that an interface is used in the resource handlers (RH) for transmitting data to and from the resource handlers, the interface being substantially independent of the application session and the resource type.

28. The method according to any of the claims 16 to 27, **characterized** in that a resource instance table (RIT) is available to each resource handler (RH), and that the resource handlers are formed to have no interspaces, wherein the resource handlers store the changes relating to data of the resource reservation instance in the resource instance table (RIT).

29. The method according to any of the claims 16 to 28, **characterized** in that a session history table (SHT) is formed for the processing step, and a resource allocation table (RAT) is formed for the resource allocation step, that the processing step and the resource allocation step are formed to have no interspaces, wherein the changes relating to data of the sessions are stored in said session history table (SHT),

and the changes relating to data of the resource allocations are stored in said resource allocation table (RAT).

30. The method according to any of the claims 16 to 29, **characterized** in that the loading situation of the processor is determined, and that the power consumption of the processor is adjusted on the basis of the loading situation as well as by scheduling of activities processed by the application sessions.

31. An electronic device (1) comprising means (2, 4) for processing application sessions, one or more processors (2), and means (2) for alternating resource reservation instances as well as the processing of application sessions being processed substantially simultaneously, **characterized** in that the application session to be processed comprises one or more application blocks (AB) in one or more application block containers (AC, AC1, AC2, AC3), and an order of processing is defined for said application blocks (AB); that the electronic device (1) comprises resource type specific resource handlers (RH) for allocating resources to the application session, resource allocation means (RAM, RAT, RH, RIT) for examining and storing a resource allocation situation, application and scheduling means (ASM) for selecting at least the application session next in turn on the basis of said allocation situation, processing means (2, ASM) for selecting and processing the application block (AB) in turn for processing in the selected application session; and the electronic device (1) is provided with a protocol (SCP) between the application and scheduling means and the processing means for determining the processing order and for implementing data transmission between said reservation means, resource allocation means, application and scheduling means, and processing means.

32. The electronic device (1) according to claim 31, **characterized** in that it is a wireless communication device.

Abstract

The present invention relates to a system comprising means (2, 4) for alternating the processing of application sessions and the necessary resource allocations in an electronic device (1) with one or more processors (2). The system comprises means (2) for alternating the resource allocations and the processing of application sessions to be processed substantially simultaneously. Of the application session to be processed, one or more application blocks (AB) are formed in one or more application block containers (AC, AC1, AC2, AC3), and a processing order is determined for said application blocks (AB). The system comprises resource type specific resource handlers (RH) for allocating resources for the application session, resource allocation means (RAM, RAT, RH, RIT) for investigating and storing the resource allocation situation, application and scheduling means (ASM) for selecting the application session next in turn for processing at least on the basis of said allocation situation, processing means (2, ASM) for selecting and processing the application block (AB) in turn for processing in the selected application session. The system is provided with a protocol (SCP) between said resource handlers, resource allocation means, application and scheduling means, and processing means, to determine the processing order and to implement data transmission between said allocation means, resource allocation means, application and scheduling means, and processing means. The invention also relates to a method for processing application sessions in an electronic device (1), as well as to an electronic device (1).

Fig. 4